



분실물 신고 서비스

**TEAM 13 – 윤영진, 반상익, 성창현, 안유성, 임수민, Rhee Benjamin Phillip**

# 목 차

<b>1. Introduction</b>	
1.1. Purpose	
1.2. Scope	
1.3. Definitions, Acronyms and Abbreviation	
1.4. References	
1.4.1 SRS 우수 사례 team13, 2023	
1.4.2 SRS 우수 사례 team9, 2023	
1.4.3 SRS 우수 사례 team10, 2022	
1.4.4 SRS 우수 사례 team 7, 2022	
1.5. Overview	
<b>2. Overall Description</b>	
2.1. Product perspective	
2.1.1. Market status	
2.2 Product Function	
2.2.1 로그인 및 회원가입	
2.2.2 정보수정	
2.2.3 분실물 등록	
2.2.4 분실물 찾기	
2.2.5 매칭	
2.2.6 유사 분실물 등록 알림	
2.2.7 매칭 후 약속 잡기	
2.2.7.1 실시간 채팅	
2.2.8 리워드 부여	
2.2.9 리워드 사용	
2.3 User Classes and Characteristic	
2.3.1 User	
2.3.2 System Manager	
2.4 Operating Environment	
2.4.1 Hardware	
2.4.2 Software	
2.5 Design and Implementation Constraints	
2.6 UserDocumentation	
2.7 Assumptions and Dependencies	
<b>3. Specific Requirements</b>	
3.1 External Interface Requirements	
3.1.1 User Interfaces	

- 3.1.2 Hardware Interfaces
- 3.1.3 Software Interfaces
- 3.1.4 Communication Interfaces
- 3.2 Functional Requirements
  - 3.2.1 Use case
    - 3.2.1.1 로그인
    - 3.2.1.2 회원 가입
    - 3.2.1.3 프로필 확인
    - 3.2.1.4 분실물 찾기
    - 3.2.1.5 분실물 등록
    - 3.2.1.6 채팅
  - 3.2.2 Use Case Diagram
  - 3.2.3 Data Dictionary
- 3.3 Non-functional Requirements
  - 3.3.1 Product Requirements
    - 3.3.1.1 Usability Requirements
    - 3.3.1.2 Efficiency Requirements
    - 3.3.1.3 Dependability Requirements
    - 3.3.1.4 Security Requirements
  - 3.3.2 Organization Requirements
    - 3.3.2.1 Environmental Requirements
    - 3.3.2.2 Development Requirements
  - 3.3.3 External Requirements
    - 3.3.3.1 Regulatory Requirements
    - 3.3.3.2 Ethical Requirements
  - 3.3.4 Organizing the Specific Requirements
    - 3.3.4.1 Context Model
- 3.4 Logical Database Requirements
- 3.5 Design Constraints
- 3.6 Standards Compliance
- 3.7 Standards Architecture
- 3.8 System Evolution
  - 3.8.1 Limitation and Assumption
  - 3.8.2 Anticipated Changes

# 1.Introduction

## 1.1 Purpose

본 문서는 성균관대학교 학생들이 잃어버린 물건을 서로 찾아주는 분실물 신고 서비스 ‘ㅇ·ㅁ·ㅈㄷ·’에 대하여 서술한다.  
 ㅇ·ㅁ·ㅈㄷ·는 2024학년도 1학기 성균관대학교  
 소프트웨어공학개론 수업의 13조 (반상익, 성장현, 안유성, 윤영진,  
 임수민, Rhee Benjamin Phillip)에 의하여 설계되고 개발될  
 예정이다.

현재 성균관대학교 교내에서 분실물이 꾸준히 발생하고 있으나,  
 분실물과 관련된 서비스들이 제대로 제공되어 있지 않다. 학교  
 홈페이지에 게시판이 개설되어 있으나 접근성이 떨어져 이용자가  
 많지 않고, 학생들의 커뮤니티인 ‘에브리타임’에서는 분실물  
 게시판이 활성화되어 있지 않다. 또한 분실물을 그 자리에 그대로  
 놔두게 되는 경우가 자주 발생하여 주인이 아닌 다른 사람들이  
 찾아가게 되는 경우도 발생한다. 따라서 13조는 이에 대한  
 해결책으로 분실물 신고 서비스인 ‘ㅇ·ㅁ·ㅈㄷ·’를 개발하기로  
 하였다.

이 문서에서는 제품, 기능, 유저, 디자인 등 요구사항 전반에 대하여  
 서술한다. 그리고 유저, 소프트웨어 인터페이스, 기능적/비기능적  
 요구사항 등 구체적인 요구사항에 대하여 설명한다.

## 1.2 Scope

분실물 신고 서비스를 제공하는 플랫폼인 ‘ㅇ·ㅁ·ㅈㄷ·’는  
 성균관대학교 교내외에서 발생한 분실물들에 대하여 주인을  
 찾아주는 것을 주 목적으로 한다. 물건을 분실한 사람, 분실물을  
 습득한 사람 모두 글을 작성할 수 있다. 글은 물품, 색상, 내용물,  
 제조사, 발견 위치, 발견 시간 등과 같은 세부 사항들을 포함하여  
 작성한다. 세부 사항들이 일치하는 게시글이 있다면 분실물을 습득한  
 사람에게 알림이 가게 된다. 이후 1대1 채팅 기능을 이용하여 시간,  
 장소를 정한 뒤 물건의 주인을 찾아줄 수 있도록 한다. 플랫폼의  
 활성화를 위해 게시글을 등록할 때 1차 리워드가 제공되고, 주인을  
 찾아줬을 때 2차 리워드가 제공된다.

## 1.3 Definitions, Acronyms and Abbreviation

아래 표는 이 문서에서 사용된 용어에 대한 정의이다.

용어	정의
사용자 (유저)	본 어플리케이션을 이용하는 사람.
분실물 신고자	분실물을 발견하여 주인을 찾아주고자 어플리케이션을 이용하는 사용자
분실물 주인	물건을 잃어버려 물건을 찾고자 어플리케이션을 이용하는 사용자

아래 표는 이 문서에서 사용된 약어에 대한 설명이다.

약어	정의
UI	User Interface
DB	Database

## 1.4 References

### 1.4.1 SRS 우수사례 1

Team 13. “성균:나누Re, 성균관대 물품 대여 platform”. SKKU, Last Modified: Sep 3 2023.

[https://github.com/skkusal/SWE3002\\_42/blob/main/2023s/Team13\\_SRS.pdf](https://github.com/skkusal/SWE3002_42/blob/main/2023s/Team13_SRS.pdf)

### 1.4.2 SRS 우수사례 2

Team 9. “Parkingo, Helps you find available parking spots on campus”. SKKU, Last Modified: Sep 3 2023.

[https://github.com/skkusal/SWE3002\\_42/blob/main/2023s/Team9\\_SRS.pdf](https://github.com/skkusal/SWE3002_42/blob/main/2023s/Team9_SRS.pdf)

### 1.4.3 SRS 우수 사례 3

Team 10. “RED:orm, Dormitory Life Assistance Platform”. SKKU, Last Modified: Sep 3 2023.

[https://github.com/skkusal/SWE3002\\_42/blob/main/2022f/Team10\\_SRS.pdf](https://github.com/skkusal/SWE3002_42/blob/main/2022f/Team10_SRS.pdf)

### 1.4.4 SRS 우수 사례 4

Team 7. “SKKU GROUP BUY Application”. SKKU, Last Modified: Sep 3 2023.

[https://github.com/skkusal/SWE3002\\_42/blob/main/2022f/Team7\\_SRS.pdf](https://github.com/skkusal/SWE3002_42/blob/main/2022f/Team7_SRS.pdf)

## 1.5 Overview

소프트웨어 요구사항 명세서는 총 3장으로 구성되어 있다.

두 번째 장은 제품이 이해관계자의 니즈와 시스템의 기반을 어떻게 충족시키는지 제품 관점에서 설명하기 위해 다양한 인터페이스, 시스템 기능과 시스템 간의 상호 연결에 대하여 기술한다. 또한 이 장은 각 이해관계자별로 시스템과의 관계에 대하여 설명한다.

세 번째 장은 상세한 용어 정의, 소프트웨어 시스템의 특징과 각 인터페이스에 대한 묘사를 제공한다. 이 장에서는 요구사항을 구체화하고 시각화하여 다양한 이해관계자가 각자의 입장에서 시스템의 흐름과 요구사항 충족 여부를 정확하게 판단할 수 있도록 돕는다.

이 문서는 소프트웨어 개발의 모든 단계에서 참조될 중요한 자료로서, 정확하고 일관된 요구사항 정의를 통해 프로젝트의 성공적인 수행을 지원한다.

## 2. Overall Description

### 2.1. Product perspective

이 플랫폼은 분실물을 찾거나 주웠을 때, 분실물을 효율적으로 돌려 주거나 받을 수 있도록 돕는 플랫폼이다. 따라서 분실물을 찾는 사람의 입장에서 자신의 물건을 찾는 과정을 간단하게 만들 수 있도록 ‘분실물 주인과 취득인 안전하고 빠른 매칭’에 집중한 플랫폼을 구현하고자 하였다. 사용자는 해당 플랫폼을 통해 분실물을 찾거나 찾아주는 과정을 빠르고 간단하게 진행할 수 있으며, 이는 분실물 회수에 소요되는 시간과 노력을 절감하는 데에 도움을 줄 수 있다. 또한 이 플랫폼은 자신의 분실물이 아님에도 물건을 취득하기 위해 악의적으로 접근하는 사용자들을 줄일 수 있는 수단 또한 고안하였다.

#### 2.1.1. Market status

현재 분실물 정보가 가장 많이 오가는 애플리케이션은 ‘에브리타임’이 있다. 이 애플리케이션은 정보 공유, 시간표 관리, 커뮤니케이션 기능 등 다양한 카테고리의 게시판을 통해 다양한 종류의 서비스를 한꺼번에 제공하는 것이 특징이다. 본 서비스에서 제공하고자 하는 분실물 관련 기능은 일반 커뮤니티에 사용자가 직업 분실물을 찾거나, 취득한 내용을 적어 게시물로 올려서 사람을 찾는 방식으로 사용되고 있다. 하지만 커뮤니티의 특성 상 새로운 다른 주제의 글들이 매우 빠르게 업로드 되고, 많은 글들 속에서 자신의 분실물에 대한 글을 찾는 것은 어려움이 있을 수 있다. 또한 불특정 다수가 글을 볼 수 있는 커뮤니티이기 때문에 실제 주인이 아니더라도 물건 취득을 목적으로 악의적으로 접근하는 사용자들을 필터링할 방법이 전무하다.

이외에 지역 기반의 커뮤니티를 구성하고 있는 ‘당근마켓’ 또한 분실물 정보를 자주 찾을 수 있는 플랫폼이다. 그러나 이 서비스 또한 상술한 ‘에브리타임’처럼 불특정 다수에게 글이 노출되는 커뮤니티의 특성을 유사하게 보유하고 있기 때문에, 같은 문제점이 존재한다.

### 2.2 Product Function

#### 2.2.1 로그인 및 회원가입

이 플랫폼을 이용하기 위한 첫 번째 단계로, 회원이 아닌 경우 회원가입 창을 통해 회원가입을 해야 하며, 회원인 경우에는 로그인을 하여 플랫폼에 접속한다. 회원가입 시에는 이메일 인증을 통해 사용자를 인증한다.

#### 2.2.2 정보수정

사용자의 패스워드, 닉네임, 이메일 등 회원가입 시 입력한 정보를 변경할 수 있는 기능이다.

### 2.2.3 분실물 등록

분실물을 취득한 사용자가 해당 분실물의 분류, 브랜드, 습득한 위치/시간 등 상세한 정보를 입력하여 분실물 정보를 등록할 수 있다.

### 2.2.4 분실물 찾기

분실물이 있는 사용자가 해당 분실물의 분류, 브랜드, 잃어버린 위치/시간 등 상세한 정보를 입력하여 분실물 정보를 등록할 수 있다. 이 때 플랫폼 내 포인트 시스템을 이용하여 사례비를 함께 첨부하여 분실물을 찾아주는 사람에게 동기부여를 해줄 수 있다.

### 2.2.5 매칭

분실물이 있는 사용자가 자신이 등록한 분실물을 터치하면, 플랫폼 내에 등록되어있는 타인이 습득한 분실물 정보에서 해당 분실물과 유사한 분실물을 검색하여 리스트를 보여준다. 이 리스트에서 사용자는 자신의 물건으로 보이는 분실물을 골라 해당 물건을 습득한 사람과 매칭을 할 수 있다.

### 2.2.6 유사 분실물 등록 알림

만약 분실물을 잃어버린 사람이 정보를 등록해놓은 시점에서, 타인이 습득한 분실물 정보 중 해당 물건과 유사도가 매우 높은 물건이 등록되었다면, 해당 분실물을 등록한 사용자에게 푸시 알림을 보내 확인을 권할 수 있도록 한다.

### 2.2.7 매칭 후 약속 잡기

#### 2.2.7.1 실시간 채팅

분실물 신고자와 분실물 주인으로 예상되는 사람이 매칭되었을 때, 분실물을 되돌려주기 위해 장소, 시간을 정하는 등의 소통을 위한 기능. 매칭된 대상과 1대1, 실시간 채팅 기능을 지원한다. 채팅방 리스트에는 채팅하고 있는 상대의 닉네임과 마지막 채팅 내용이 있으며 클릭하면 해당 유저와의 채팅방으로 연결된다.

### 2.2.8 리워드 부여

사용자는 분실물을 등록할 때에 사례금으로 사용할 수 있는 포인트를 충전할 수 있다.

### 2.2.9 리워드 사용

자신이 분실물을 찾아준 답례로 습득한 포인트를 프린터기와 같은 교내의 작은 서비스들에서 활용할 수 있도록 할 예정이다.



## 2.3. User Classes and Characteristic

### 2.3.1. User

이 플랫폼의 사용자는 성균관대학교에 재학중인 학생들로 제한한다. 그리고 사용자는 기존에 물건을 잃어버린 경우, 일반 커뮤니티를 통해 물건을 찾는 데 어려움을 겪었던 사람이라고 가정한다. 사용자는 한국어를 사용할 수 있다고 가정하며, 자신의 기기에 애플리케이션을 설치하고 사용할 수 있는 기본지식이 있다고 가정한다.

### 2.3.2. System Manager

이 플랫폼의 시스템 관리자는 시스템 전문가로 제한된다. 이들은 본 플랫폼의 시스템에 대한 충분한 지식이 있어야 하고 전반적인 시스템 상황을 이해할 수 있어야 한다. 또한 네트워크에 대한 충분한 지식이 있어야 하며, 실제 사람들이 소유한 재화나 포인트가 오가는 시스템이기에 도덕적, 윤리적 의식이 있어야 한다.

## 2.4. Operating Environment

### 2.4.1. Hardware

해당 플랫폼은 **Android** 기반 운영 체제의 모바일 디바이스를 위해 설계되었다. 본 플랫폼에 접속하는 사용자 기기는 최소 **512MB**의 메모리 잔여 용량과 **300MB** 이상의 하드디스크 잔여 용량이 있음을 전제한다.

### 2.4.2. Software

해당 플랫폼은 최소 안드로이드 **14** 버전 이상의 모바일 디바이스에서 사용되는 것을 전제하고 있다. 따라서 다른 **OS**가 설치된 기기나 낮은 안드로이드 버전에서는 서비스를 원활하게 이용할 수 없을 수 있다.

## 2.5 Design and Implementation Constraints

해당 플랫폼은 다음 체크리스트를 고려하여 디자인하고 구현될 예정이다.

- A. 시스템은 사용자의 편의와 요구사항을 고려해야 하며, 해당 내용에 따라 수정이 가능하다.
- B. 시스템은 꼭 필요한 경우가 아니라면 가급적 무료로 제공되는 소프트웨어를 사용한다.
- C. 가능한 범위 내에서 오픈소스 소프트웨어를 적극 활용한다.
- D. 한국어가 필요하다
- E. 최소 **Kotlin v2.0.0** 이상, **React v18.2.0** 이상, **Flask v2.2.5** 이상, **Python v3.7** 이상을 사용해 개발한다
- F. **Android 14** 이상의 OS가 설치된 모바일 디바이스에서 테스트한다.
- G. 시스템은 **512MB** 이상의 **RAM** 을 가진 디바이스에서 실행되어야 한다.
- H. 시스템 설치와 구동을 위해 최소 **300MB** 의 여유공간이 디바이스에 존재해야 한다.
- I. 유저 디바이스와 서버는 **REST API** 구조 기반으로 **HTTP** 프로토콜로 통신한다.
- J. 시스템은 **Android** 운영체제에서 실행 가능하다.

## 2.6. User Documentation

사용자가 해당 플랫폼을 잘 활용할 수 있도록 돕기 위해 몇 가지 문서가 필요하다. 첫째, 하드웨어 및 소프트웨어 요구사항을 다룬 문서가 필요하며, 둘째 사용자 설명서가 필요하다. 사용자 설명서의 경우 각 탭은 어떤 목적을 위해 개설되었는지, 분실물을 등록할 때 어떤 정보들을 어떤 양식으로 적어야 하는지에 대한 내용을 포함해야 한다. 또한 물건 등록 정보와 매칭/채팅 내역에 대한 예시가 담겨있는 스크린샷이 포함되어야 한다. 마지막으로 시스템 개발자의 연락처를 제공하여 기타 버그나 문제점을 보고할 수 있도록 한다

## 2.7. Assumption and Dependencies

본 문서에서 구현할 플랫폼은 최소 **Android 14** 이상의 운영체제를 사용하는 모바일 디바이스에서 사용된다는 가정하에 작성되었다. 더 낮은 버전의 **Android** 운영체제의 디바이스에서도 접근하여 사용할 수 있다. 그러나 모든 버전의 정상 작동을 보장하지 않는다.

## 3. Specific Requirements

### 3.1. External Interface Requirements

#### 3.1.1 User Interface

The login interface features a light gray background. At the top, the text '아맛다' is displayed in a large, bold, black font. Below this, there are two white input fields with rounded corners. The first field is labeled '아이디를 입력해주세요' and the second is labeled '비밀번호를 입력해주세요'. A prominent pink button labeled '로그인' is positioned below the password field. At the bottom, there is a link '아직 회원이 아니라면?' followed by a gray button labeled '회원가입'.

[Figure 1] 로그인 화면의 user interface

1. 사용자의 아이디와 비밀번호를 입력한다.
2. 로그인 버튼을 누를 시 메인 화면 [Figure 3]로 넘어간다.
3. 회원이 아닐 시 회원가입 버튼을 누르면 [Figure 2]로 넘어간다.

The registration interface has a light gray background. It contains several input sections: '아이디 입력' with a white field and a hint '알파벳, 숫자, 3글자 이상'; '비밀번호 입력' with a white field and a hint '알파벳, 숫자, 8자 이상', followed by a '비밀번호 확인' field; '이메일 입력' with a white field and an '인증번호 입력' field next to it, with a pink '인증' button; and '닉네임 입력' with a white field and a hint '알파벳, 한글, 숫자 2글자 이상'. A large pink button labeled '회원가입' is at the bottom.

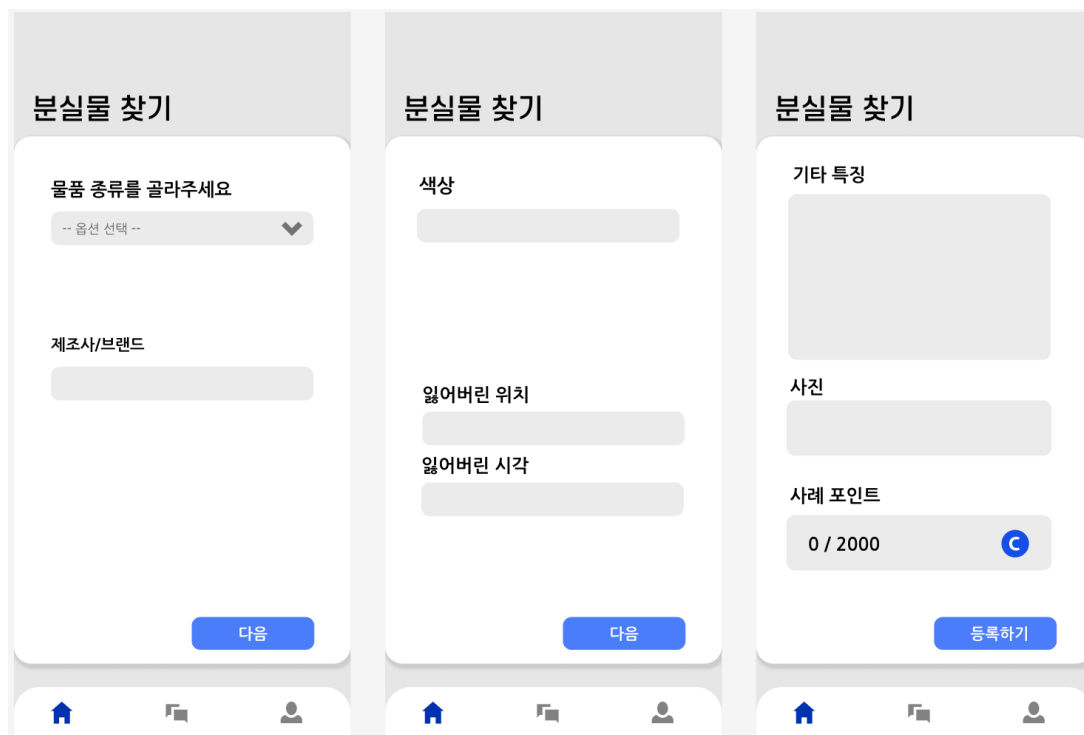
[Figure 2] 회원 가입의 user interface

1. 사용할 아이디와 비밀번호, 닉네임을 입력한다.
2. 이메일을 이용하여 사용자 인증을 한다.
3. 회원가입 버튼을 누르면 Figure 1으로 넘어간다.



[Figure 3] 메인 화면의 user interface

1. 화면의 가운데에는 분실물 찾기/분실물 등록 버튼을 만들어 해당 기능에 접속할 수 있다.
2. 하단에는 내비게이션 바가 있어 왼쪽부터 메인 화면, 채팅방 리스트 화면, 마이페이지 화면에 해당하는 버튼이 있어 각각을 누르면 해당하는 페이지로 이동한다.



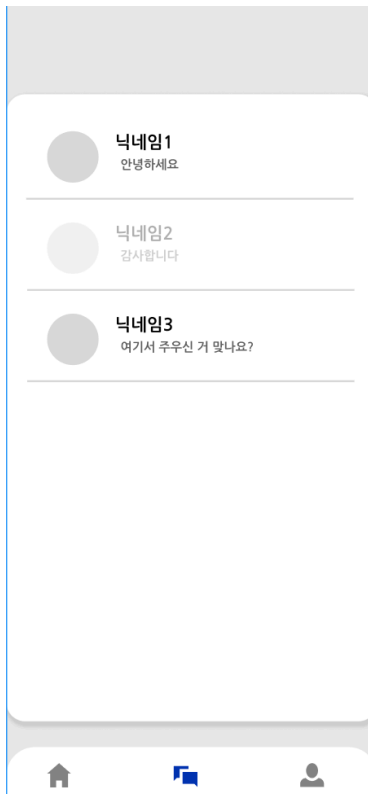
[Figure 4] 분실물 찾기 정보 등록 과정의 user interface

1. 가장 첫 페이지에는 상단부터 잃어버린 물건의 종류를 고를 수 있는 옵션과, 제조사나 브랜드를 입력할 수 있는 텍스트 박스가 있다.
2. 우측 하단의 다음 버튼을 누르면 다음 페이지로 이동한다.

3. 다음 페이지에서는 상단부터 물건의 색상을 입력하는 텍스트 박스와 잃어버린 위치와 시각을 입력하는 박스가 존재한다.
4. 우측 하단의 다음 버튼을 누르면 다음 페이지로 이동한다.
5. 상단부터 기타 사용자가 명시하고 싶은 물건의 특징을 입력하는 텍스트 박스와 물건의 사진을 업로드할 수 있는 박스, 필요 시 사례금을 앱 내 포인트의 형태로 추가할 수 있는 박스가 있다.
6. 우측 하단의 다음 버튼을 누르면 분실물을 찾는 정보가 앱에 등록된다.

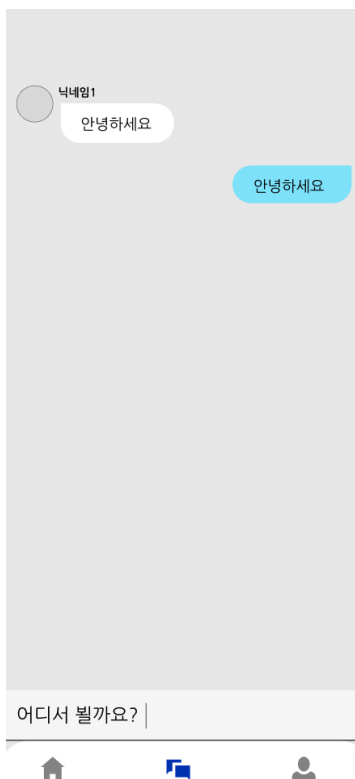
[Figure 5] 분실물 등록 정보 등록 과정의 user interface

1. 가장 첫 페이지에는 상단부터 잃어버린 물건의 종류를 고를 수 있는 옵션과, 제조사나 브랜드를 입력할 수 있는 텍스트 박스가 있다.
2. 우측 하단의 다음 버튼을 누르면 다음 페이지로 이동한다.
3. 다음 페이지에서는 상단부터 물건의 색상을 입력하는 텍스트 박스와 습득한 위치와 시각을 입력하는 박스가 존재한다.
4. 우측 하단의 다음 버튼을 누르면 다음 페이지로 이동한다.
5. 상단부터 기타 사용자가 명시하고 싶은 물건의 특징을 입력하는 텍스트 박스와 물건의 사진을 업로드할 수 있는 박스, 필요 시 사례금을 앱 내 포인트의 형태로 추가할 수 있는 박스가 있다.
6. 우측 하단의 다음 버튼을 누르면 분실물을 찾는 정보가 앱에 등록된다.



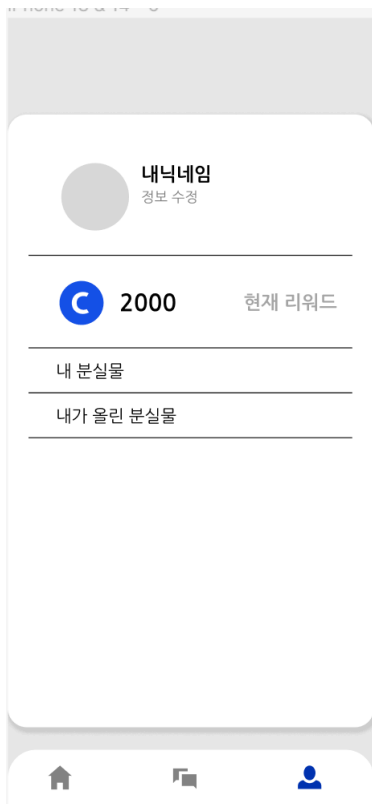
[Figure 6] 채팅 리스트 페이지의 user interface

1. 분실물을 위해 연결된 사람들과의 채팅 리스트를 제공한다. 각 아이템에는 상대방의 닉네임과 마지막 대화를 표시한다.
2. 각 리스트의 아이템을 누르면 해당 채팅 페이지로 이동한다.
3. 리스트 아이템이 반투명해지게 만들어 닫힌 채팅방을 표시한다.



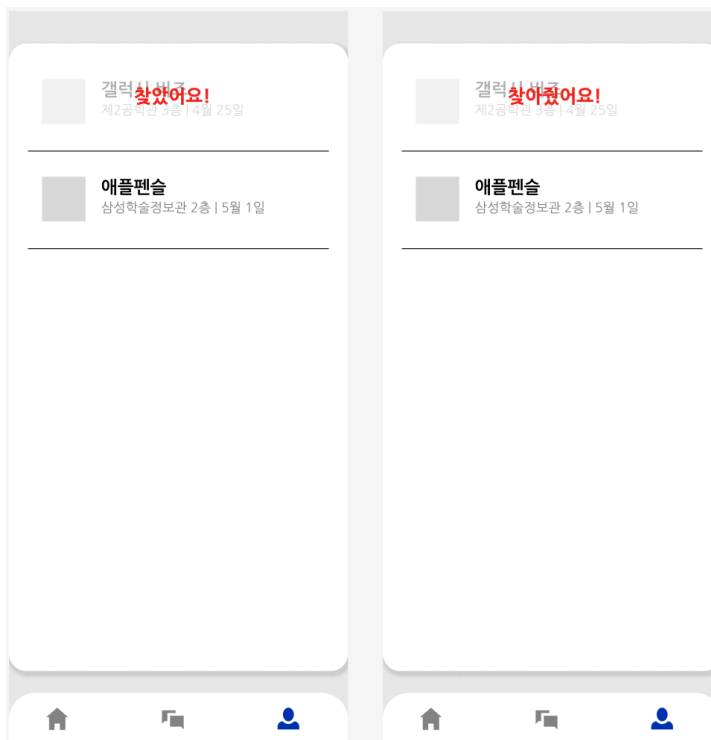
[Figure 7] 채팅 페이지 user interface

1. 상세한 채팅 내역을 확인할 수 있다.
2. 하단의 텍스트 박스에 내용을 입력하여 채팅을 보낼 수 있다.



[Figure 8] 마이페이지 user interface

1. 상단에 사용자의 프로필 사진과 닉네임을 표시한다.
2. 정보 수정 버튼을 누를 시 정보 수정 페이지로 이동한다.
3. 사용자가 현재 보유하고 있는 앱 내 포인트를 표시한다. 누를 경우 포인트 관리 페이지로 이동한다.
4. '내 분실물'을 누르면 사용자가 올렸던 분실물 찾기 정보를 볼 수 있는 페이지로 이동한다.
5. '내가 올린 분실물'을 누르면 사용자가 올렸던 분실물 등록 정보를 볼 수 있는 페이지로 이동한다.



[Figure 8-1, 8-2] 내 분실물/내가 올린 분실물 페이지

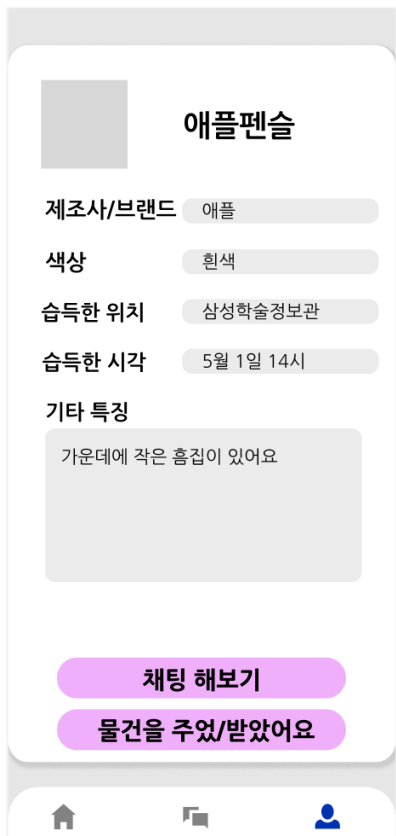
1. 사용자가 올린 분실물 찾기/분실물 등록 정보의 리스트를 제공한다.
2. 각 리스트 아이템에는 상품의 사진, 이름, 잃어버린 장소와 시간을 표시한다.
3. 내 분실물 리스트의 아이템을 누르는 경우 등록된 분실물 중 해당 아이템과 비슷한 물건을 조회하는 페이지로 이동한다.
4. 내가 등록한 분실물 리스트에서 아이템을 누르는 경우 상세 정보 페이지로 이동한다.





[Figure 9] 내 분실물과 비슷한 분실물 조회 페이지 user interface

1. 상단에 자신의 분실물에 대한 사진, 이름, 잃어버린 위치와 시간 정보를 제공한다.
2. 그 아래에 해당 분실물일 확률이 높은 분실물을 조회하여 보여준다.
3. 리스트의 아이템을 누를 시 상세 정보 페이지로 이동한다.



[Figure 10] 분실물 상세 정보 페이지

1. 해당 분실물에 대한 제조사/색상/습득위치/습득장소 등 상세 정보를 보여준다.
2. 하단의 '채팅 해보기' 버튼을 누르면 해당 분실물을 주워 등록한 사람과의 채팅 페이지로 이동한다.
3. 분실물 거래가 끝난 후 하단의 '물건을 주었/받았어요' 버튼을 누르면 해당 분실물 정보가 사라지고, 필요 시 포인트가 이동한다.

### 3.1.2. Hardware Interface

- 64bit CPU, 512MB of RAM, 200MB of drive space 이상의 하드웨어 사양을 가진 모바일 및 PC 로 접속해야 한다.
- Oracle Cloud Infrastructure 프리티어 서버를 활용해 개발을 진행한다.
- 사용되는 인스턴스인 VM.Standard.E2.1.Micro의 경우 아래 표와 같은 리소스를 제공한다.

OCPU	대역폭 (Gbps)	메모리 (GB)	스토리지
1	0.48	1	Block storage only

### 3.1.3 Software Interfaces

Kotlin(v2.0.0), React(v18.2.0), Flask(v2.2.5), Python(v3.7.3)을 사용해 개발되었다.

사용자는 Android 14 이상 버전의 안드로이드 OS가 설치된 기기를 이용해 접속해야 한다.

### 3.1.4 Communication Interfaces

분실물 신고자와 분실물 주인 간, 즉 User와 User간의 분실물 전달을 위해 장소와 시간을 정하는 Communication. User Device와 User Device간의 통신을 지원하며 Socket 통신을 통해 실시간 양방향 통신을 지원한다.

## 3.2 Function Requirements

### 3.2.1 Use case

#### 3.2.1.1 로그인

Use case name	로그인 / 로그아웃
Actor	회원 가입이 끝난 유저
Description	메인 페이지의 로그인 / 로그아웃 기능을 사용해 로그인 / 로그아웃 한다
Normal Course	1. ID, password를 입력해 로그인을 시도한다. 2. ID, password 가 올바르지 않을 경우 재시도 한다.

	3. 로그아웃 버튼을 눌러 로그아웃 한다.
Pre-Condition	네트워크 사용이 가능해야 한다.
Post-Condition	로그인한 유저는 로그아웃 하기 전까지 세션에 유지되어야한다.
Assumption	

### 3.2.1.2 회원 가입

Use case name	회원 가입 및 프로필 생성
Actor	새로 가입하려는 유저
Description	서비스를 사용하기 위해 회원 가입하고 프로필을 생성한다.
Normal Course	<ol style="list-style-type: none"> <li>1. 새로 가입하고자 하는 유저는 회원가입 버튼을 눌러 회원가입 페이지로 이동한다.</li> <li>2. 유저는 ID, password, 이메일 등 유저의 정보를 입력한다.</li> <li>3. ID 중복 여부를 체크하고 중복 된다면 재입력을 요청한다.</li> <li>4. 성균관대학교 학생임을 인증하기 위해 학교 이메일을 입력하고 인증번호로 인증을 받는다.</li> <li>5. 회원가입을 완료한다.</li> </ol>
Pre-Condition	<ol style="list-style-type: none"> <li>1. 네트워크 사용이 가능해야 한다.</li> <li>2. 성균관대학교 메일(****@skku.edu ) 이용이 가능한 유저여야 한다.</li> <li>3. 플랫폼에 가입한 이력이 없는 유저여야 한다.</li> </ol>
Post-Condition	<ol style="list-style-type: none"> <li>1. Password는 암호화되어 노출되지 않아야 한다.</li> <li>2. 메일로 보낸 인증번호와 유저가 입력한 인증번호가 일치하는지 확인할 수 있어야 한다.</li> </ol>

	3. 유저의 정보는 안전하게 보호되어야 한다
Assumption	

### 3.2.1.3 프로필 확인

Use case name	프로필 확인
Actor	프로필을 확인하려는 유저
Description	자신의 프로필을 확인한다.
Normal Course	<ol style="list-style-type: none"> <li>1. 네비게이션 바의 내 정보를 클릭하여 자신의 프로필을 확인한다.</li> <li>2. 내 닉네임, 현재 리워드, 내 분실물, 내가 올린 분실물을 확인할 수 있다.</li> <li>3. 내 분실물을 클릭하여 내가 올린 분실물 찾기 게시글을 확인할 수 있다.</li> <li>4. 내가 올린 분실물을 클릭하여 내가 올린 분실물 등록 게시글을 확인할 수 있다.</li> </ol>
Pre-Condition	네트워크 사용이 가능해야 한다.
Post-Condition	유저의 정보는 안전하게 보호되어야 한다.
Assumption	

### 3.2.1.4 분실물 찾기

Use case name	분실물 찾기
Actor	찾으려는 분실물 정보를 등록하려는 유저
Description	분실물을 찾기 위해 분실물 정보를 등록한다.
Normal Course	<ol style="list-style-type: none"> <li>1. 분실물을 찾고자하는 유저는 분실물 찾기 버튼을 눌러 분실물 정보 입력 페이지로 이동한다.</li> </ol>

	<p>2. 유저는 분실물 종류, 제조사, 색상, 잃어버린 위치, 장소, 특징, 사진, 사례 포인트 정보를 입력하고 등록하기 버튼으로 찾으려는 분실물을 등록한다.</p> <p>3. 분실물 종류는 필수 입력값이므로 입력하지 않고 다음 페이지로 넘어갈 수 없다.</p> <p>4. 작성자는 자신의 분실물 찾기 게시글을 수정/삭제할 수 있다.</p>
Pre-Condition	네트워크 사용이 가능해야 한다.
Post-Condition	<p>1. 분실물 찾기의 분실물 특징과 분실물 등록의 분실물 특징이 일치할 경우 해당 분실물 등록의 정보를 열람할 수 있다.</p> <p>2. 입력된 사례 포인트는 분실물을 찾을 경우 분실물 제공자에게 제공된다.</p>
Assumption	

### 3.2.1.5 분실물 등록

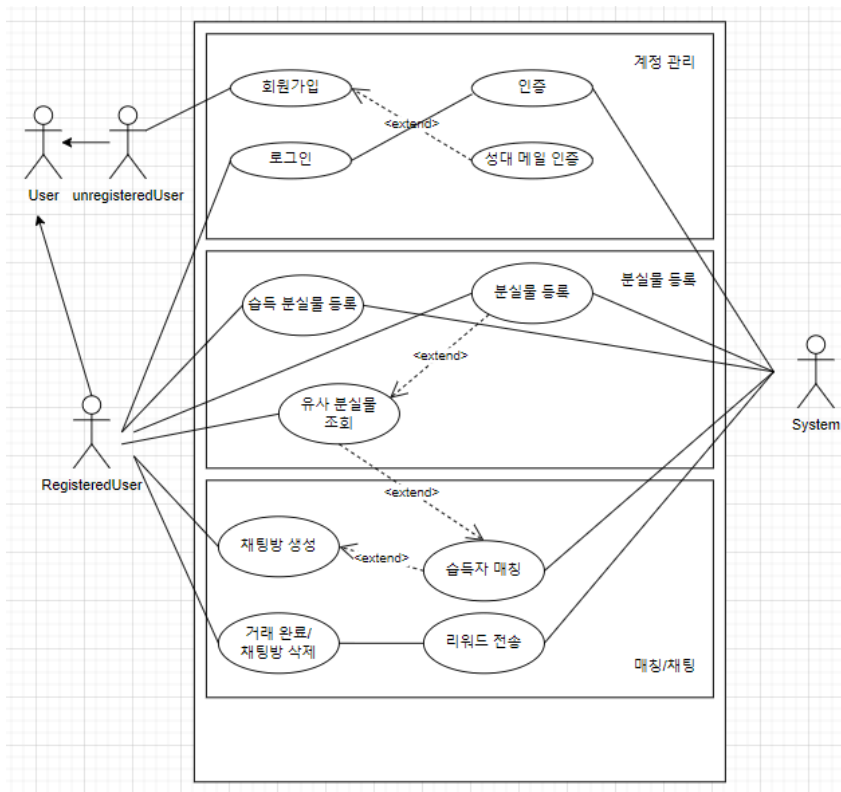
Use case name	분실물 등록
Actor	분실물을 습득하여 분실물 정보를 등록하려는 유저
Description	분실물을 습득하여 분실물 정보를 등록한다.
Normal Course	<p>1. 분실물을 습득한 유저는 분실물 등록 버튼을 눌러 분실물 정보 입력 페이지로 이동한다.</p> <p>2. 유저는 분실물 종류, 제조사, 색상, 습득한 위치, 장소, 특징, 사진을 입력하고 등록하기 버튼으로 습득한 분실물의 정보를 등록한다.</p> <p>3. 분실물 종류는 필수 입력값이므로 입력하지 않고 다음 페이지로 넘어갈 수 없다.</p> <p>4. 작성자는 자신의 분실물 등록 게시글을 수정/삭제할 수 있다.</p>
Pre-Condition	네트워크 사용이 가능해야 한다.

Post-Condition	분실물 등록의 분실물 특징과 분실물 찾기의 분실물 특징이 일치할 경우 해당 분실물 찾기의 정보를 열람할 수 있다.
Assumption	

### 3.2.1.6 채팅

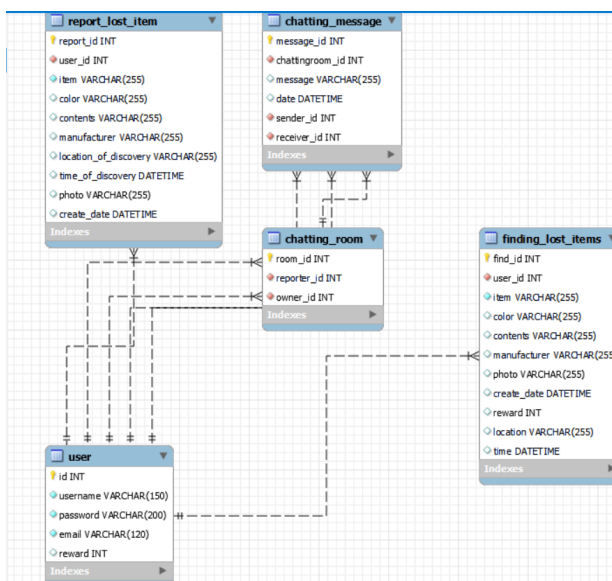
Use case name	채팅
Actor	분실물 신고자, 분실물 주인
Description	분실물 신고자와 분실물 주인은 매칭된 상대와 실시간으로 채팅을 할 수 있다.
Normal Course	내용 입력란에 채팅 내용을 입력한다. 전송 버튼을 클릭해 채팅 내용을 상대방에게 전송한다.
Pre-Condition	신고자가 신고한 분실물과 분실물 주인이 찾으려고 하는 분실물의 특징이 일치하는 경우 신고자와 분실물 주인이 매칭되어 채팅방에 입장한다.
Post-Condition	보낸 채팅은 채팅방에 기록된다.
Assumption	

### 3.2.2 Use-case Diagram



[Figure 11] Use-case Diagram

### 3.2.3 Data Dictionary



[Figure 12] DB diagram

### 3.3 Non-functional Requirements

#### 3.3.1 Product Requirements

##### 3.3.1.1 Usability Requirements

- 사용자가 별도의 메뉴얼을 보지 않더라도 편리하게 사이트를 이용할 수 있도록 가시성이 좋은 UI 구성과 직관적인 아이콘을 사용하여야 한다.
- 페이지의 텍스트와 아이콘만으로 각 기능을 파악할 수 있어야 한다.
- 채팅에 사용되는 언어는 한국어와 영어를 지원한다.

##### 3.3.1.2 Efficiency Requirements

- 상대방에게 채팅을 전송했을 때 2초 내에 전송이 완료되어야 하고, 서비스 내에서의 유저의 행동에 대한 응답 속도는 4초를 넘지 말아야 한다.

##### 3.3.1.3 Dependability Requirements

- 사용자가 ID, password를 정확하게 입력했을 경우 항상 해당 유저의 ID에 맞는 계정으로 로그인할 수 있어야 한다.
- 분실물 정보를 등록하는 경우 유저가 입력한 정보를 손상 없이 전달할 수 있어야 한다.
- 매칭된 유저와 채팅을 진행할 경우 유저가 입력한 채팅의 내용이 온전하게 상대방에게 전달되어야 한다.
- 채팅 메시지는 보낸 순서에 맞게 누락 없이 채팅방에 기록되어야 한다.
- 서버 내 이슈로 인한 서비스의 장애를 최소화하기 위해 서버 유지/보수를 진행해야 한다.

##### 3.3.1.4 Security Requirements

- 유저는 회원 가입시 성균관대학교 메일 ( \*\*\*\*\*@skku.edu/g.skku.edu )을 통해 성균관대학교 학생임을 인증 받아야 하고 성균관대학교 학생임이 인증된 유저만 사이트를 사용할 수 있다
- 사용자가 회원가입한 이후 사용자의 개인정보는 안전하게 보호되어야 하며 비밀번호의 경우 암호화되어 관리자도 확인할 수 없어야 한다
- 유저 간의 채팅 내용은 매칭된 분실물 신고자와 분실물 주인 이외에 다른 유저는 볼 수 없어야 한다.



### 3.3.2 Organization Requirements

#### 3.3.2.1 Environmental Requirements

- 해당 플랫폼은 기본적으로 **Kotlin v2.0.0**을 기반으로 개발되며 **Android 14** 버전 이상이 설치된 모바일 디바이스에서 사용할 수 있도록 개발되어야 한다.,
- **Android 14** 이상의 **OS**가 설치된 모바일 디바이스에서 테스트한다.
- 시스템은 **512MB** 이상의 **RAM** 을 가진 디바이스에서 실행되어야 한다.
- 시스템 설치와 구동을 위해 최소 **300MB** 의 여유공간이 디바이스에 존재해야 한다.

#### 3.3.2.2 Development Requirements

- **UI/UX**의 전체적인 디자인은 **Figma**를 통해 진행되어야 하며 아이콘, 텍스트를 직관적으로 이해할 수 있도록 사용한다.
- **Kotlin, React**를 이용해 **Webview** 기반의 안드로이드 앱 개발을 진행하도록 한다.
- **Flask**를 사용해 **Back End** 개발을 진행하도록 한다.
- 데이터베이스는 **ID, password** 등의 사용자 정보를 관리하며 각 데이터들은 사용자의 요청이 있지 않는 한 변경되지 않아야 한다.

### 3.3.3 External Requirements

#### 3.3.3.1 Regulatory Requirements

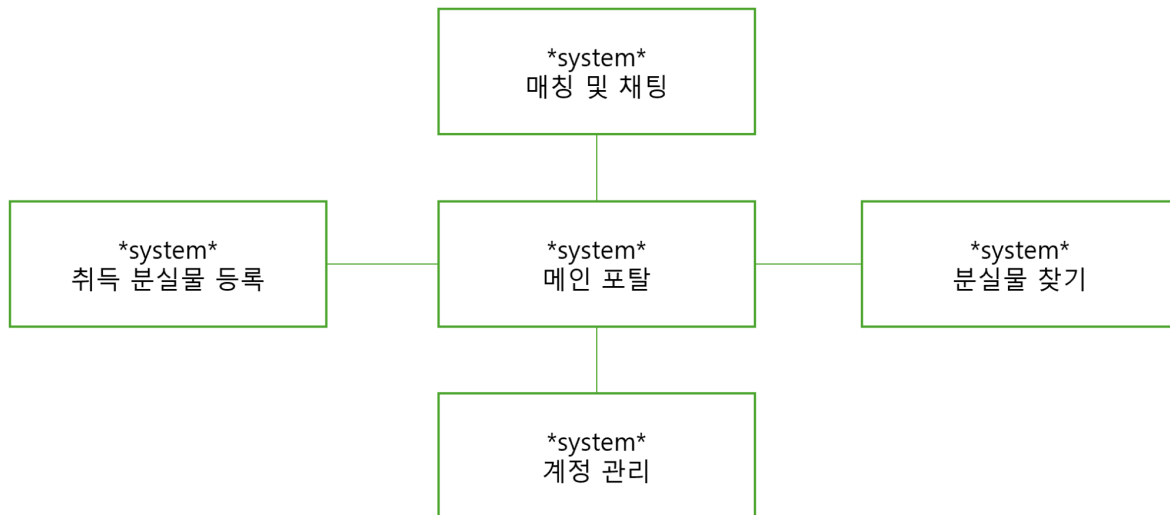
- 본 명세서에서 사용하는 오픈소스 프로그램들의 사용 규정에 맞게 시스템이 개발되어야 한다.
- 개발에 있어 사용하는 시스템들은 저작권 보호법을 위반해서는 안 되며 원작자의 요청 시 이를 삭제해야 한다.

#### 3.3.3.2. Ethical Requirements

- 해당 플랫폼을 운영함에 있어서 부적절한 방법을 통한 금전적 이득을 취하지 않아야 한다.
- 시스템 사용자가 플랫폼 내 포인트에 대해 부적절한 접근을 할 수 없도록 차단해야 한다.

### 3.3.4 Organizing the Specific Requirements

#### 3.3.4.1 .Context model



[Figure13] Context model

## 3.4 Logical Database Requirements

MySQL 를 통해 유저 정보가 담긴 데이터베이스를 관리한다. 시스템은 ID, password 등 사용자의 정보와 등록된 분실물 정보, 채팅 등을 데이터베이스에 저장한다. 게시물, 채팅 등을 생성할 때 이는 각 계정에 맞게 권한이 귀속되어야 한다. 데이터베이스는 정보를 처리하기 위한 기본 성능을 가질 수 있도록 관리되어야 한다.

#### 3.4.1.1 Design Constraints

- Android 운영 체제를 사용하는 모바일 디바이스에서 접근이 가능해야 한다. 시스템은 PostgreSQL (v16)의 데이터베이스를 사용하도록 설계되어야 한다.
- HTTP 통신이 가능하도록 설계되어야 한다.
- Android/Webview 개발을 위해 nodejs v20.11.1 이상, React v18.12.0 이상, Kotlin v2.0.0 이상을 사용하에 한다.
- 채팅은 소켓 통신을 통해 실시간으로 이루어지도록 설계되어야 한다.
- Backend 개발을 위해 Python v3.7 이상, Flask v2.2.5 이상을 사용해야 한다.

### 3.4.1.2 Standard Compliance

- 웹 기반 시스템 관리 도구는 HTML 표준을 따른다. 프로그램 버전 관리 시스템으로는 git 및 github을 사용한다.
- 시스템 개발은 Windows, macOS, Linux 모두 가능하며, 64bit CPU, 512MB 이상의 메모리 용량, 300MB 이상의 여유 공간이 디바이스에 존재하여야 한다.

## 3.5. System Evolution

### 3.5.1.1 Limitation and Assumption

본 시스템에서는 사용자들이 학교 메일을 통해서만 성균관대학교 학생 인증을 받을 수 있도록 설계되었다. 따라서 학교 메일을 사용할 수 없는 학생들은 없다고 가정한다. 만약 학교 메일 사용이 불가능한 학생이 존재할 경우 이 시스템을 사용할 수 없다는 한계가 있다. 또한 기존 커뮤니티의 분실물 게시물에서 문제가 되었던 악의적인 사용자의 접근을 줄이기 위해 사용자가 등록한 정보에 해당하는 분실물의 정보만 사용자에게 노출시키지만, 이 방법 또한 악의적인 사용자의 접근을 원천 차단하지는 못한다는 한계점을 가지고 있다.

### 3.5.1.2 Anticipated Changes

본 시스템이 위에 설명된 한계점을 극복하기 위해서는 학교 메일을 통한 학생 인증 이외의 다른 인증 절차를 추가해야 한다. 또한 분실물이 악의적인 사용자에게 전달되었을 경우 이를 시스템에 신고하여 해당 사용자를 차단하거나, 해당 피해에 대한 법적 조치를 지원하는 시스템을 구현한다면 더욱 신뢰도 높은 서비스가 될 수 있을 것이다.

이러한 변화가 이루어진다면 사용자는 해당 서비스에 대해 더 깊은 신뢰를 할 수 있고, 신뢰를 기반으로 더 많은 분실물 거래가 이루어질 수 있고, 이는 서비스의 본래의 목적을 달성할 수 있을 것이다.